

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Yuji YAMAMOTO Atty. Docket No. 107156-00194

Serial No.: New application Examiner: Not Assigned

Filed: July 22, 2003 Art Unit: Not Assigned

For: LOGARITHMIC TRANSFORMER AND METHOD OF LOGARITHMIC TRANSFORMATION

CLAIM FOR PRIORITY

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313

July 22, 2003

Sir:

The benefit of the filing dates of the following prior foreign applications in the following foreign country is hereby requested for the above-identified patent application and the priority provided in 35 U.S.C. §119 is hereby claimed:

Japanese Patent Application No. 2002-214046 filed on July 23, 2002

In support of this claim, a certified copy of said original foreign application is filed herewith.

It is requested that the file of this application be marked to indicate that the requirements of 35 U.S.C. §119 have been fulfilled and that the Patent and Trademark Office kindly acknowledge receipt of these documents.

Please charge any fee deficiency or credit any overpayment with respect to this paper to Deposit Account No. 01-2300.

Respectfully submitted,

George E. Oram, Jr.
Registration No. 27,931

Customer No. 004372

AREN'T FOX KINTNER PLOTKIN & KAHN, PLLC

1050 Connecticut Avenue, N.W., Suite 400

Washington, D.C. 20036-5339

Washington, D.C. 20585

Fax: (202) 638-4810

(translation)

PATENT OFFICE
JAPANESE GOVERNMENT

This is to certify that the annexed is a true copy of
the following application as filed with this office.

Date of application: July 23, 2002

Application Number: Japanese Patent Application
No. 2002-214046
[ST.10/C] : [JP2002-214046]

Applicant(s): **Pioneer Corporation**

Date of this certificate: January 10, 2003

Commissioner,
Japan Patent Office

Certificate No. 2002-3104924

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2002年 7月23日

出願番号

Application Number:

特願2002-214046

[ST.10/C]:

[JP2002-214046]

出願人

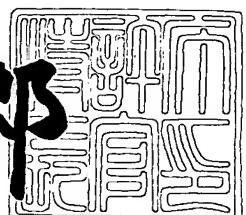
Applicant(s):

バイオニア株式会社

2003年 1月10日

特許庁長官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2002-3104924

【書類名】 特許願
【整理番号】 56P0682
【提出日】 平成14年 7月23日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 7/00
G06F 7/556

【発明者】

【住所又は居所】 埼玉県川越市山田字西町25番地1 パイオニア株式会社 川越工場内

【氏名】 山本 雄治

【特許出願人】

【識別番号】 000005016

【氏名又は名称】 パイオニア株式会社

【代理人】

【識別番号】 100063565

【弁理士】

【氏名又は名称】 小橋 信淳

【選任した代理人】

【識別番号】 100118898

【弁理士】

【氏名又は名称】 小橋 立昌

【手数料の表示】

【予納台帳番号】 011659

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 対数変換器及び対数変換方法

【特許請求の範囲】

【請求項1】 対数変換すべきデジタルデータの各ビットのうち論理“1”となっている最も上位に位置するビットの位置を表す2進数のバイナリデータを生成する対数変換上位ビット列生成手段と、

前記最も上位に位置するビットより下位に位置するビット列を前記デジタルデータより求める対数変換下位ビット列生成手段とを備え、

前記2進数のバイナリデータを前記デジタルデータを対数変換した場合に得られる整数の対数変換値、前記下位に位置するビット列を前記デジタルデータを対数変換した場合に得られる小数点以下の対数変換値、とする対数変換データとして出力することを特徴とする対数変換器。

【請求項2】 前記対数変換上位ビット列生成手段は、

前記対数変換すべきデジタルデータの各ビットのうち論理“1”となっている最も上位に位置するビットの位置を検出する検出部と、

前記検出部の検出結果に従って、前記2進数のバイナリデータの各ビット列を生成する生成部と、を備えることを特徴とする請求項1に記載の対数変換器。

【請求項3】 前記検出部は、

前記対数変換すべきデジタルデータをデコードすることにより、前記論理“1”となっている最も上位に位置するビットを検出することを特徴とする請求項2に記載の対数変換器。

【請求項4】 前記生成部は、

前記対数変換すべきデジタルデータの各ビットの位置を表す2進数のバイナリデータを有し、前記検出部の前記検出結果に従って、前記各ビットの位置を表す2進数のバイナリデータの1つを選択することにより、前記論理“1”となっている最も上位に位置するビットの位置を表す前記2進数のバイナリデータを生成することを特徴とする請求項2に記載の対数変換器。

【請求項5】 前記生成部は、前記検出結果に従って、前記各ビットの位置を表す2進数のバイナリデータの1つを選択する切換え回路で形成されているこ

とを特徴とする請求項4に記載の対数変換器。

【請求項6】 前記検出部は、前記対数変換すべきデジタルデータの各ビットのうち、前記論理“1”となっている最も上位に位置するビットより下位に位置するビット列を対象外とするデータを出力する第1の論理回路と、

前記対数変換すべきデジタルデータの各ビットのうち、前記データで対象外とされたビットを除くことで、前記論理“1”となっている最も上位に位置するビットを検出する第2の論理回路と、を備えることを特徴とする請求項2に記載の対数変換器。

【請求項7】 前記第1の論理回路は、

前記対数変換すべきデジタルデータの最上位ビットから最下位ビットまでの各ビットを入力すると共に、夫々の出力と入力とが前記最上位ビットから最下位ビット側へと対応付けて従属接続された複数の論理和ゲートを備え、

前記各論理和ゲートが、上位ビット側の論理和ゲートの出力と前記デジタルデータの各ビットとを論理和演算することにより、前記対象外とするデータを生成することを特徴とする請求項6に記載の対数変換器。

【請求項8】 第2の論理回路は、前記対数変換すべきデジタルデータの各ビットと、前記対象外とするデータとを論理積演算することにより、前記論理“1”となっている最も上位に位置するビットを検出することを特徴とする請求項6に記載の対数変換器。

【請求項9】 前記下位ビット列生成手段は、

前記デジタルデータの各ビットのうち、前記論理“1”となっている最も上位に位置するビットの次の下位ビットを含む予め決められた所定個数のビット列を、前記小数点以下の対数変換値として求めることを特徴とする請求項1に記載の対数変換器。

【請求項10】 前記下位ビット列生成手段は、前記最も上位に位置するビットの次の下位ビットから前記デジタルデータの最下位ビットまでのビット数が前記所定個数に満たないと、満たない分のビットを前記最下位ビットに追加して、所定個数のビット列を生成することを特徴とする請求項9に記載の対数変換器。

【請求項11】 前記下位ビット列生成手段は、

前記対数変換すべきデジタルデータを、連続した前記所定個数ずつのビット列の組として入力する下位ビット列抽出部と、

前記デジタルデータの前記論理“1”となっている最も上位に位置するビットの次の下位ビットを最上位ビットとして有する前記所定個数のビット列を、前記小数点以下の対数変換値として、前記下位ビット列抽出部に抽出させる下位ビット列選択部と、を備えることを特徴とする請求項9又は10に記載の対数変換器。

【請求項12】 前記下位ビット列選択部は、

前記対数変換すべきデジタルデータをデコードすることにより、前記論理“1”となっている最も上位に位置するビットを検出することを特徴とする請求項11に記載の対数変換器。

【請求項13】 前記下位ビット列抽出部は、

前記下位ビット列選択部でデコードされた前記検出結果に従って、最も上位に位置するビットの次の下位ビットを最上位ビットとして有する前記所定個数のビット列を抽出することを特徴とする請求項12に記載の対数変換器。

【請求項14】 前記下位ビット列抽出部は、前記下位ビット列選択部の前記検出結果に従って、前記所定個数のビット列を抽出する切換え回路で形成されていることを特徴とする請求項11に記載の対数変換器。

【請求項15】 対数変換すべきデジタルデータの各ビットのうち論理“1”となっている最も上位に位置するビットの位置を表す2進数のバイナリデータを生成する上位ビット列生成工程と、

前記最も上位に位置するビットより下位に位置するビット列を前記デジタルデータより求める下位ビット列生成工程とを備え、

前記2進数のバイナリデータを前記デジタルデータを対数変換した場合に得られる整数の対数変換値、前記下位に位置するビット列を前記デジタルデータを対数変換した場合に得られる小数点以下の対数変換値とすることで、前記デジタルデータに対する対数変換データを生成することを特徴とする対数変換方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、例えばデジタルデータ等を対数変換する対数変換器及び対数変換方法に関する。

【0002】

【従来の技術】

従来、デジタル信号処理の分野において、例えばビット数の多いデジタルデータをそれより少ないビット数のデジタルデータにデータ圧縮したり、複数のデジタルデータを乗算又は除算する代わりに、対数変換を施したデジタルデータを加算又は減算することにより演算の効率化を図る等、有用な手段として対数変換が広く利用されている。

【0003】

従来の一般的な対数変換器として、図1 (a) に示す構成のものが知られている。

【0004】

この対数変換器は、対数変換を高速で行うべくハードウェアで構成され、同図 (a) に示すように、論理回路としてのアドレスデコーダ1と、読み出し専用の半導体メモリで形成された対数ROM2を備えて構成されている。

【0005】

対数ROM2は、所謂ルックアップテーブルの形態で形成されており、対数変換すべきデジタルデータに対応した多数の対数変換データが予め記憶されている。

【0006】

アドレスデコーダ1は、対数変換すべきデジタルデータが入力されると、そのデジタルデータをアドレスデータに変換（デコード）して対数ROM2のアドレスを指定することにより、デジタルデータに対応する対数変換データを読み出させる。

【0007】

つまり、対数変換すべきデジタルデータの値と、対数ROM2のアドレスを指

定するためのアドレスデータと、そのアドレスデータで指定されるアドレスに予め記憶されている対数変換データとが互いに一対一の関係で決められているため、対数変換すべきデジタルデータをアドレスデコーダ1に供給すると、そのデジタルデータに対応する対数変換データを対数変換の結果として、対数ROM2から出力させ、それによって高速の対数変換を行うことを可能にしている。

【0008】

【課題を解決するための手段】

ところが、従来の対数変換器は、回路規模が極めて大きなものとなってしまうという問題があった。

【0009】

例えば高密度実装や小形軽量化等の必要な電子機器に対数変換器を設ける場合や、対数変換器を半導体集積回路装置として形成する場合等、その他の場合において、解決すべき重要な課題となっていた。

【0010】

具体的な一事例を上げて説明すれば、図1(a)に示したアドレスデコーダ1は、図1(b)に示すような複数段の論理回路1a, 1b等で構成され、更に各論理回路1a, 1b等は多数の論理演算素子(インバータ、ORゲート、NORゲート、ANDゲート、NANDゲート等)を組み合わせて形成されている。

【0011】

ここで、初段に位置する複数の論理回路1a, 1a…は、入力されるnビット(nは任意の自然数)のデジタルデータを複雑に組み合わせて論理演算するための複数個のインバータとNORゲートで形成され、次段の論理回路1bも同様に、複数の論理回路1a, 1a…から出力される多数のデータを複雑に組み合わせて論理演算すべく多数の論理演算素子で形成され、更に論理回路1bは1段のみではなく多段の論理回路で形成されている。

【0012】

このように、対数ROM2の全アドレス空間をアクセスし得るアドレスデータを生成しなければならない等の必要性から、アドレスデコーダ1の回路規模が大きくなっていた。

【0013】

また、対数ROM2がルックアップテーブルの形態で形成されていることからも、回路規模が大きくなる一因となっていた。

【0014】

つまり、対数ROM2には、対数変換すべきデジタルデータのビット数nに応じて、少なくとも 2^n 通りの対数変換データを予め記憶しておく必要がある。

【0015】

更に各対数変換データを、nビットのデジタルデータよりも少ないビット数mのデータとして記憶させたとしても、対数ROM2の総記憶容量として少なくとも $2^n \times m$ ビット分が必要である。

【0016】

こうしたことから、記憶容量の大きな対数ROM2が必要となり、回路規模が大きくなる一因となっていた。

【0017】

本発明は、例示したような従来の課題に鑑みてなされたものであり、新規な構成を有する対数変換器を提供することを目的とする。

【0018】

【課題を解決するための手段】

請求項1に記載の対数変換器は、対数変換すべきデジタルデータの各ビットのうち論理“1”となっている最も上位に位置するビットの位置を表す2進数のバイナリデータを生成する対数変換上位ビット列生成手段と、前記最も上位に位置するビットより下位に位置するビット列を前記デジタルデータより求める対数変換下位ビット列生成手段とを備え、前記2進数のバイナリデータを前記デジタルデータを対数変換した場合に得られる整数の対数変換値、前記下位に位置するビット列を前記デジタルデータを対数変換した場合に得られる小数点以下の対数変換値、とする対数変換データとして出力することを特徴とする。

【0019】

請求項2に記載の対数変換器は、請求項1に記載の対数変換器において、前記対数変換上位ビット列生成手段は、前記対数変換すべきデジタルデータの各ビッ

トのうち論理“1”となっている最も上位に位置するビットの位置を検出する検出部と、前記検出部の検出結果に従って、前記2進数のバイナリデータの各ビット列を生成する生成部とを備えることを特徴とする。

【0020】

請求項3に記載の対数変換器は、請求項2に記載の対数変換器において、前記検出部は、前記対数変換すべきデジタルデータをデコードすることにより、前記論理“1”となっている最も上位に位置するビットを検出することを特徴とする。

【0021】

請求項4に記載の対数変換器は、請求項2に記載の対数変換器において、前記生成部は、前記対数変換すべきデジタルデータの各ビットの位置を表す2進数のバイナリデータを有し、前記検出部の前記検出結果に従って、前記各ビットの位置を表す2進数のバイナリデータの1つを選択することにより、前記論理“1”となっている最も上位に位置するビットの位置を表す前記2進数のバイナリデータを生成することを特徴とする。

【0022】

請求項5に記載の対数変換器は、請求項4に記載の対数変換器において、前記生成部は、前記検出結果に従って、前記各ビットの位置を表す2進数のバイナリデータの1つを選択する切換え回路で形成されていることを特徴とする。

【0023】

請求項6に記載の対数変換器は、請求項2に記載の対数変換器において、前記検出部は、前記対数変換すべきデジタルデータの各ビットのうち、前記論理“1”となっている最も上位に位置するビットより下位に位置するビット列を対象外とするデータを出力する第1の論理回路と、前記対数変換すべきデジタルデータの各ビットのうち、前記データで対象外とされたビットを除くことで、前記論理“1”となっている最も上位に位置するビットを検出する第2の論理回路とを備えることを特徴とする。

【0024】

請求項7に記載の対数変換器は、請求項6に記載の対数変換器において、前記

第1の論理回路は、前記対数変換すべきデジタルデータの最上位ビットから最下位ビットまでの各ビットを入力すると共に、夫々の出力と入力とが前記最上位ビットから最下位ビット側へと対応付けて従属接続された複数の論理和ゲートを備え、前記各論理和ゲートが、上位ビット側の論理和ゲートの出力と前記デジタルデータの各ビットとを論理和演算することにより、前記対象外とするデータを生成することを特徴とする。

【0025】

請求項8に記載の対数変換器は、請求項6に記載の対数変換器において、第2の論理回路は、前記対数変換すべきデジタルデータの各ビットと、前記対象外とするデータとを論理積演算することにより、前記論理“1”となっている最も上位に位置するビットを検出することを特徴とする。

【0026】

請求項9に記載の対数変換器は、請求項1に記載の対数変換器において、前記下位ビット列生成手段は、前記デジタルデータの各ビットのうち、前記論理“1”となっている最も上位に位置するビットの次の下位ビットを含む予め決められた所定個数のビット列を、前記小数点以下の対数変換値として求めることを特徴とする。

【0027】

請求項10に記載の対数変換器は、請求項9に記載の対数変換器において、前記下位ビット列生成手段は、前記最も上位に位置するビットの次の下位ビットから前記デジタルデータの最下位ビットまでのビット数が前記所定個数に満たないと、満たないと分のビットを前記最下位ビットに追加して、所定個数のビット列を生成することを特徴とする。

【0028】

請求項11に記載の対数変換器は、請求項9又は10に記載の対数変換器において、前記下位ビット列生成手段は、前記対数変換すべきデジタルデータを、連続した前記所定個数ずつのビット列の組として入力する下位ビット列抽出部と、前記デジタルデータの前記論理“1”となっている最も上位に位置するビットの次の下位ビットを最上位ビットとして有する前記所定個数のビット列を、前記

小数点以下の対数変換値として、前記下位ビット列抽出部に抽出させる下位ビット列選択部と、を備えることを特徴とする。

【0029】

請求項12に記載の対数変換器は、請求項11に記載の対数変換器において、前記下位ビット列選択部は、前記対数変換すべきデジタルデータをデコードすることにより、前記論理“1”となっている最も上位に位置するビットを検出することを特徴とする。

【0030】

請求項13に記載の対数変換器は、請求項12に記載の対数変換器において、前記下位ビット列抽出部は、前記下位ビット列選択部でデコードされた前記検出結果に従って、最も上位に位置するビットの次の下位ビットを最上位ビットとして有する前記所定個数のビット列を抽出することを特徴とする。

【0031】

請求項14に記載の対数変換器は、請求項11に記載の対数変換器において、前記下位ビット列抽出部は、前記下位ビット列選択部の前記検出結果に従って、前記所定個数のビット列を抽出する切換え回路で形成されていることを特徴とする。

【0032】

請求項15に記載の対数変換方法は、対数変換すべきデジタルデータの各ビットのうち論理“1”となっている最も上位に位置するビットの位置を表す2進数のバイナリデータを生成する上位ビット列生成工程と、前記最も上位に位置するビットより下位に位置するビット列を前記デジタルデータより求める下位ビット列生成工程とを備え、前記2進数のバイナリデータを前記デジタルデータを対数変換した場合に得られる整数の対数変換値、前記下位に位置するビット列を前記デジタルデータを対数変換した場合に得られる小数点以下の対数変換値として、前記デジタルデータに対する対数変換データを生成することを特徴とする。

【0033】

【発明の実施の形態】

本発明の好適な実施の形態を図2乃至図4を参照して説明する。

なお、図2乃至図4は、本実施形態の対数変換器の構成並びに機能を示した図である。

【0034】

また、説明の便宜上、対数変換すべきデジタルデータBは、一般的表現形態であるB($b_{n-1}, b_{n-2}, b_{n-3}, \dots, b_2, b_1, b_0$)等で表されるnビットのバイナリデータとし、また、対数変換後の対数変換データDも同様に、D($d_{m-1}, d_{m-2}, d_{m-3}, \dots, d_{m-p}, d_{m-p-1}, \dots, d_2, d_1, d_0$)等で表し得るmビットのバイナリデータであるものとして説明することとする。

【0035】

また、上述の各データBとDのビット数nとmは、必ずしもこれに限定されるものではないが、一般に行われる対数変換の意義を考慮して、 $n \geq m$ の関係にあるものとして説明することとする。

【0036】

図2において、本対数変換器は、対数変換すべき入力データBに対して対数変換の処理を施し、対数変換の結果としての対数変換データDを出力する。

【0037】

かかる対数変換の処理を行うべく、対数変換上位ビット列生成部3と対数変換下位ビット列生成部4が備えられている。

【0038】

対数変換上位ビット列生成部3は、図3に示す如く、アクティブビット検出部3aと上位ビット列生成部3bを備えて構成されている。

【0039】

対数変換下位ビット列生成部4は、図4に示す如く、下位ビット選択部4aと下位ビット列抽出部4bを備えて構成されている。

【0040】

ただし、これらの対数変換上位ビット列生成部3と対数変換下位ビット列生成部4及び、アクティブビット検出部3a、上位ビット列生成部3b、下位ビット

選択部4a、下位ビット列抽出部4bは、個別独立して形成することが可能であるが、融合した1又は複数の回路等として形成することも可能である。

【0041】

つまり、これらの各構成要素3, 4, 3a, 3b, 4a, 4bを最適に論理設計する際、個別独立な構成とする代わりに、例えば論理演算素子や一部回路等の共有化を図ることで、融合した1又は複数の回路等として回路規模をシミュリンクすることが可能である。

【0042】

図2、図3において、対数変換上位ビット列生成部3は、nビットの変換すべき入力データBが供給されると、その入力データBの各ビット $b_{n-1}, b_{n-2}, b_{n-3}, \dots, b_2, b_1, b_0$ のうち、論理“1”となっているビットであって最も上位に位置しているビット（以下「アクティブビット」という）を検出する。

【0043】

この検出処理を、図3のアクティブビット検出部3aが行い、上述の検出した最上位のビットをアクティブビットとし、更にそのアクティブビットが入力データBにおける何ビット目の位置にあるかを表すためのビット番号Sを特定する。

【0044】

例えば、上述のアクティブビットが最上位（MSB）からk番目に位置するビット b_{n-k} であった場合、そのビット b_{n-k} のビット番号Sをn-kとする

【0045】

更に具体的な場合を例示すれば、入力データBが、B(0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1)で表される16ビット（n=16）のデータであった場合、最上位ビット（MSB） b_{15} から4番目に位置する論理“1”的ビット b_{n-k} （つまり、 b_{12} ）をアクティブビットとし、更に、ビット番号Sを「12」とする。

【0046】

なお、ビット b_{12} よりも下位のビット $b_{11} \sim b_0$ で、論理“1”となって

いるものがあっても、アクティブビットとはしない。

【0047】

そして、アクティブビット検出部3aは、ビット番号Sをアクティブビットの検出結果として、上位ビット列生成部3bと対数変換下位ビット列生成部4へ供給する。

【0048】

次に、上位ビット列生成部3bは、mビットの対数変換データDにおける、最上位ビット d_{m-1} を含む上位p個分のビットである、対数変換上位ビット列 $D_{UP}(d_{m-1}, d_{m-2}, d_{m-3}, \dots, d_{m-p})$ を生成する。

【0049】

ここで、対数変換上位ビット列 D_{UP} のビット数pは、nビットの入力データBに対応させて、 $n = 2^p$ の関係（別言すれば、 $p = \log_2 n$ の関係）を満足するように予め決められている。

【0050】

そして、アクティブビット検出部3aからビット番号Sが供給されると、そのビット番号Sに相当する2進数のバイナリデータ（p個のビットから成るバイナリデータ）を生成し、そのビット数pのバイナリデータを対数変換上位ビット列 D_{UP} とする。

【0051】

より具体的な事例を上げて説明すれば、入力データBが、前述したB(0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1)で表される16ビットのデータであった場合、アクティブビットのビット番号Sは「12」となるから、ビット番号Sに相当する2進数のバイナリデータ(1, 1, 0, 0)を生成する。

【0052】

そして、 $n = 2^p$ の関係から、16ビット($n = 16$)の入力データBに対応する対数変換上位ビット列 D_{UP} のビット数pは「4」となり、更に上述のビット番号Sに相当する2進数のバイナリデータ(1, 1, 0, 0)となることから、上述の4ビットの対数変換上位ビット列を $D_{UP}(1, 1, 0, 0)$ とする。

【0053】

また、生成すべき対数変換データDのビット数mが、入力データBのビット数nより小さい場合であっても、 $n = 2^p$ の関係から対数変換上位ビット列 D_{UP} のビット数pを決め、ビット番号Sに相当する2進数のバイナリデータに相当する対数変換上位ビット列 D_{UP} 、つまりビット数pの対数変換上位ビット列 D_{UP} を生成する。

【0054】

このように、アクティブビット検出部3aと上位ビット列生成部3bを備えた対数変換上位ビット列生成部3は、入力データB中の論理“1”となっているビットのうち、最も上位に位置しているビットをアクティブビットとして着目し、アクティブビットのビット番号Sの2進数のバイナリデータを対数変換上位ビット列 D_{UP} として生成する。

【0055】

次に、図2、図4に基づいて対数変換下位ビット列生成部4を説明する。

【0056】

対数変換下位ビット列生成部4は、nビットの変換すべき入力データBが供給されると、その入力データBに基づいて、対数変換データDの下位 $m-p$ ビット分を表す対数変換下位ビット列 D_{LOW} を生成する。

【0057】

この対数変換下位ビット列 D_{LOW} を生成するのに、図4の下位ビット選択部4aと下位ビット列抽出部4bが備えられている。

【0058】

下位ビット選択部4aは、入力データBの各ビット $b_{n-1}, b_{n-2}, b_n-3, \dots, b_2, b_1, b_0$ のうち、上述のビット番号Sで特定されたアクティブビット b_{n-k} よりも下位に位置するビット列($b_{n-k-1}, b_{n-k-2}, \dots, b_1, b_0$)を選択し、その選択結果を下位ビット列抽出部4bへ出力する。

【0059】

なお、本実施形態では、ビット番号Sに基づいて、下位のビット列(b_{n-k}

$b_{n-k-2}, \dots, b_1, b_0$ を選択することとしているが、他の方法で行っててもよい。

【0060】

変形例として、上述の p ビットの対数変換上位ビット列 $D_{UP} (d_{m-1}, d_{m-2}, \dots, d_{m-p})$ に基づいて、下位のビット列 $(b_{n-k-1}, b_{n-k-2}, \dots, b_1, b_0)$ を選択してもよい。つまり、対数変換上位ビット列 $D_{UP} (d_{m-1}, d_{m-2}, \dots, d_{m-p})$ はビット番号 S の 2 進数のデータであることから、ビット番号 S の代わりに対数変換上位ビット列 $D_{UP} (d_{m-1}, d_{m-2}, \dots, d_{m-p})$ に基づいて、下位のビット列 $(b_{n-k-1}, b_{n-k-2}, \dots, b_1, b_0)$ を選択するようにしてもよい。

【0061】

次に、下位ビット列抽出部 4 b は、生成すべき対数変換データ D のビット数 m と対数変換上位ビット列 D_{UP} のビット数 p との差 $(m-p)$ に相当する個数 q のビットを、下位のビット列 $(b_{n-k-1}, b_{n-k-2}, \dots, b_1, b_0)$ のうちの上位から抽出し、その抽出した q 個（つまり、 $m-p$ 個）のビット列 $(b_{n-k-1}, b_{n-k-2}, \dots, b_{n-k-1-q})$ を、対数変換データ D における対数変換下位ビット列 $D_{LOW} (d_{m-p-1}, d_{m-p-2}, \dots, d_1, d_0)$ とする。

【0062】

なお、下位のビット列 $(b_{n-k-1}, b_{n-k-2}, \dots, b_1, b_0)$ のビット総数が、 q 個（つまり、 $m-p$ 個）よりも少ない場合には、不足分のビットを論理 “0” として、最下位（LSB）ビット b_0 に続けて付加することにより、 q 個のビットから成る対数変換下位ビット列 $D_{LOW} (d_{m-p-1}, d_{m-p-2}, \dots, d_1, d_0)$ を生成する。

【0063】

そして、図 2 に示すように、 p ビットの対数変換上位ビット列 $D_{UP} (d_{m-1}, d_{m-2}, d_{m-3}, \dots, d_{m-p})$ を、入力データ B を対数変換することによって得られる整数値として出力し、 q ビットの対数変換下位ビット列 $D_{LOW} (d_{m-p-1}, \dots, d_2, d_1, d_0)$ を、入力データ B を対数変換するこ

とによって得られる小数点以下の値として出力する。

【0064】

そして、これら対数変換上位ビット列 D_{UP} を対数変換データ D の上位ビット列、対数変換下位ビット列 D_{LOW} を対数変換データ D の下位ビット列とする総計 m ビットの対数変換データ D を出力する。

【0065】

より具体的な場合を例示すると、下位ビット列抽出部4bは次に述べる処理を行うこととなる。なお、一具体例として、16ビット($n=16$)の入力データ B に対して8ビット($m=8$)の対数変換データ D を生成すべく本対数変換器を形成し、入力データ B が、前述の $B(0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1)$ で表される16ビットのデータであった場合について説明することとする。

【0066】

入力データ $B(0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1)$ が入力されると、論理“1”となる最上位ビットは第12ビット b_{12} であることから、既述したように対数変換上位ビット列 $D_{UP}(d_{m-1}, d_{m-2}, \dots, d_{m-p})$ は、 $D_{UP}(1, 1, 0, 0)$ となる。

【0067】

ここで、生成すべき8ビット($m=8$)の対数変換データ D のうち、対数変換上位ビット列 $D_{UP}(1, 1, 0, 0)$ が4ビット($p=4$)分を占めることとなり、残りの4ビット($m-p=4$)に対数変換下位ビット列 D_{LOW} を割り当てる。

【0068】

そして、対数変換下位ビット列 D_{LOW} の候補として、入力データ $B(0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1)$ のうち、上述の第12ビット b_{12} よりも下位のビット列($b_{11}, b_{10}, b_9, b_8, \dots, b_0$)であるビット列($0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1$)を選択し、更に、上位4ビット分のビット列(b_{11}, b_{10}, b_9, b_8)に該当するビット列($0, 1, 1, 0$)を対数変換下位ビット列 $D_{LOW}(d_3,$

d_2, d_1, d_0 ）、すなわち $D_{LOW}(0, 1, 1, 0)$ とする。

【0069】

そして、対数変換上位ビット列 $D_{UP}(1, 1, 0, 0)$ と対数変換下位ビット列 $D_{LOW}(0, 1, 1, 0)$ とによって、最終的な対数変換データ $D(1, 1, 0, 0, 0, 1, 1, 0)$ とする。

【0070】

なお、図2中、変換後の対数変換データ D に付けられているポインタ t が、小数点の位置であることを示している。これにより、例えば対数変換データ D をレジスタ回路等でバッファリングしてから並列出力する場合に、そのレジスタ回路等における物理的なビットの位置をポインタ t で表すことで、対数変換上位ビット列 D_{UP} と対数変換下位ビット列 D_{LOW} との境（小数点の位置）を表すことが可能である。

【0071】

ただし、ポインタ t によって、対数変換データ D における対数変換上位ビット列 D_{UP} と対数変換下位ビット列 D_{LOW} との境（小数点の位置）を表す表現方法に限定されるものではない。

【0072】

変形例として、対数変換上位ビット列 D_{UP} と対数変換下位ビット列 D_{LOW} との境の位置に1ビット追加し、その追加した1ビットを小数点の位置を表すビットとして、総計 $m+1$ ビットの対数変換データ D としてもよい。

【0073】

また、他の変形例として、対数変換上位ビット列 D_{UP} と対数変換下位ビット列 D_{LOW} との境の位置に1ビット追加し、その追加した1ビットを小数点の位置を表すビットとともに、対数変換下位ビット列 D_{LOW} のうちの最下位ビット d_0 を除去することにより、総計 m ビットの対数変換データ D とするようにしてもよい。

【0074】

本実施形態によれば、入力データ B を対数変換した場合、その対数変換データ D の値は、理論値とほぼ一致する結果が得られた。なお、具体的な対数変換の精

度については、次の実施例において説明するが、図12の評価結果に示すように、本対数変換器で16ビットの入力データBを対数変換した場合、その対数変換データDの値は、理論値とほぼ一致する結果が得られた。

【0075】

更に、本実施形態の対数変換器によれば、図1に示した従来のアドレスデコーダ1に比して、極めて少ない論理演算素子で形成することができ、回路規模の低減等を実現することができる。

【0076】

例えば、対数変換上位ビット列生成部3は、論理演算素子を用いた小規模の比較器やデコーダ回路によって、入力データB中のアクティブビット及びビット番号Sを検出し特定することができる。

【0077】

また、ビット番号Sに相当するpビットの対数変換上位ビット列D_{UP}を生成するのに、論理演算素子を用いた小規模のデコーダ回路によって、ビット番号Sからバイナリデータを生成することができる。

【0078】

また、対数変換下位ビット列生成部4は、入力データBから所定のビットを単に選択して抽出することで、対数変換下位ビット列D_{LOW}を求めることができるため、規模の小さな例えばレジスタ回路等で実現することができる。

【0079】

また、従来技術のような対数変換に相当するデータをROM等に予め記憶しておく構成ではないことから、回路規模を低減することができる。

【0080】

以上例示したように、本実施形態の対数変換器は回路規模の低減等を可能にする新規な構成を有し、デジタル信号処理の分野において有用な効果を提供することができるものである。

【0081】

【実施例】

次に、図5乃至図12を参照して、より具体的な実施例を説明する。

【0082】

なお、図5は、本実施例の対数変換器を利用したFM／AM受信機の構成を示す図、図6及び図7は、本対数変換器の回路を示す図、図8乃至図11は、本対数変換器の動作及び機能を示す図、図12は本対数変換器の評価結果を示す図である。

【0083】

図5において、本実施例の対数変換器を利用したFM／AM受信機は、到来電波をアンテナ5を介して受信するRFアンプ6と、PLL及びVCO等の回路を有して局発信号を生成する局発回路8と、局発信号と受信信号とを混合して中間周波信号（IF信号）等を生成するミキサ7と、中間周波信号を増幅するIFアンプ9が備えられている。

【0084】

更に、IFアンプ9にて増幅された中間周波信号をデジタルデータにアナログデジタル変換するA／D変換部10と、そのデジタルデータに対して所定のデジタル信号処理を施すことによってFM検波信号を検波するFM検波部11と、そのデジタルデータに対して所定のデジタル信号処理を施すことによってAM検波信号を検波するAM検波部12が備えられている。

【0085】

ここで、AM検波部12は、上述のAM検波信号を生成する他、A／D変換部10から供給されるデジタルデータを全波整流し、AM検波を行うことによって、送信局の有無等を検出するのに利用される電界強度を示すデータを出力する。

【0086】

ただし、上述の全波整流とAM検波によって電界強度を示すデータを生成すると、その電界強度を示すデータはリニアなデータとなり、本FM／AM受信機のダイナミックレンジ等を考慮すると、リニアなデータのままでは処理し難いことから、電界強度を示すデータを対数変換して処理すべく、本実施例の対数変換器が設けられている。

【0087】

次に、図6及び図7に基づいて、本実施例の対数変換器の回路構成を詳述する

【0088】

本対数変換器は、16ビット($n=16$)の電界強度を示すリニアなデータ(以下「入力データ」という)Bを、8ビット($m=8$)の対数変換データDに変換する構成となっており、図6に示す回路が、図2に示した対数変換上位ビット列生成部3、図7に示す回路が、図2に示した対数変換下位ビット列生成部4となっている。

【0089】

図6において、対数変換上位ビット列生成部3は、15個の論理和ゲート(ORゲート)U₁₄~U₀と、15個の論理積ゲート(ANDゲート)G₁₄~G₀と、マルチプレクサ13を備えて構成されている。

【0090】

マルチプレクサ13は、図中の符号dで示す17個の入力端子を備えたデータ入力ポートと、図中の符号eで示す17個の制御端子eを備えた切換え制御ポートとを有し、更に図中の符号d₆, d₅, d₄で示す3ビットのバイナリデータを並列出力する出力ポートが備えられている。

【0091】

データ入力ポートの各入力端子dには、図示するように、10進数の「0」から「7」までの各値に相当する3ビットのバイナリデータ(0, 0, 0)~(1, 1, 1)が予め印加されている。

【0092】

つまり、図中の最上位に位置する入力端子(後述のデコードビットg₁₅が印加される制御端子eと対を成している入力端子)dにはバイナリデータ(1, 1, 1)、次の入力端子dにはバイナリデータ(1, 1, 0)、更に次の入力端子dにはバイナリデータ(1, 0, 1)が印加され、残余の各入力端子dにも、図示の如く、3ビットのバイナリデータ(0, 0, 0)~(1, 1, 1)のうちの所定のバイナリデータが予め印加されている。

【0093】

また、これら3ビットのバイナリデータ(0, 0, 0)~(1, 1, 1)の各

論理値“1”と“0”は、いわゆる電源供給端子とグランド端子に、プルアップ抵抗とプルダウン抵抗を接続するという簡単な回路構成によって設定されている。

【0094】

上述の各制御端子eには、後述するデコードビット $g_{15} \sim g_0$ と h_0 が夫々独立して供給されるようになっている。

【0095】

そして、図中、隣り合って示されている1個ずつの入力端子dと制御端子eが1組ずつの対を成しており、各組における制御端子eに論理“1”となるデコードビットが印加されると、その制御端子eと対を成している入力端子dに印加されている上述のバイナリデータを出力データ(d_6, d_5, d_4)として、出力ポートより出力する。

【0096】

つまり、詳細については後述するが、マルチプレクサ13は、各制御端子eに印加されるデコードビット $g_{15} \sim g_0, h_0$ の何れかが論理“1”になると、その論理“1”に設定された制御端子eと対を成している入力端子dに印加されているバイナリデータを出力データ(d_6, d_5, d_4)とすべくマルチルックスする。

【0097】

次に、ORゲート $U_{14} \sim U_0$ とANDゲート $G_{14} \sim G_0$ は、所定の関係で組み合わせて配線されることにより、最上位ビット(MSB) b_{15} から最下位ビット LSB b_0 までの16ビットの入力データB($b_{15}, b_{14}, \dots, b_1, b_0$)を16ビットのデコードデータG($g_{15}, g_{14}, \dots, g_1, g_0$)にデコードすると共に、1ビット分のデコードビット h_0 をデコードし、上述の17個の制御端子eへ並列出力する。

【0098】

ただし、デコードデータGのうちの最上位のデコードビット g_{15} については、上述のORゲートとANDゲートが関与することなく、入力データBの最上位のビット b_{15} をそのままマルチプレクサ13の最上位に位置する制御端子eへ

供給するようになっている。

【0099】

また、2入力型の各ANDゲート $G_{14} \sim G_0$ は、一方が論理反転入力端子、他方が論理非反転入力端子となっており、ORゲート $U_{14} \sim U_0$ は、夫々2入力型の論理和ゲートで形成され、そしてORゲート $U_{14} \sim U_0$ 及びANDゲート $G_{14} \sim G_0$ は、入力データBの各ビット $b_{15}, b_{14}, \dots, b_1, b_0$ に対応して設けられている。

【0100】

そして、図示するように、ORゲート $U_{14} \sim U_0$ は、夫々の上位に位置するORゲートの出力と、入力データB ($b_{15}, b_{14}, \dots, b_1, b_0$) の所定のビットとの論理和演算を行うべく、上位側から下位側へといわゆる従属接続が成されており、各ORゲート $U_{14} \sim U_1$ の出力 $u_{14} \sim u_1$ がANDゲート $G_{13} \sim G_0$ の上記の論理反転入力端子に夫々供給され、ANDゲート $G_{14} \sim G_0$ より出力されるデコードビット $g_{14} \sim g_0$ がマルチプレクサ13の所定の制御端子eに供給されるように配線されている。

【0101】

そして、マルチプレクサ13の出力ポートより出力される3ビットのビット列 (d_6, d_5, d_4) と、ORゲート U_8 の出力 u_8 を最上位ビット d_7 とする総計4ビットのビット列 (d_7, d_6, d_5, d_4) を、上述の対数変換上位ビット列 D_{UP} として出力する。

【0102】

次にかかる構成を有する対数変換上位ビット列生成部3の動作を説明する。

【0103】

まず、ORゲート $U_{14} \sim U_0$ の動作を説明する。ORゲート $U_{14} \sim U_0$ は、入力データBの各ビット $b_{15} \sim b_0$ のうちで、論理“1”となっている最上位のビットに対応しているORゲート及び、そのORゲートより下位に位置しているORゲートの出力を全て論理“1”にすべく動作する。

【0104】

例えば、入力データB (b_{15}, \dots, b_0) の各ビット列が上述した (0, 0

, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1) であった場合、論理 “1” となっている最上位のビット b_{12} に対応しているORゲート U_{12} に論理 “1” の値が印加され、更にORゲート U_{12} より下位に位置しているORゲート $U_{11} \sim U_0$ は従属接続されていることから、これらのORゲート $U_{11} \sim U_0$ にも論理 “1” の値が印加される。

【0105】

この結果、ORゲート $U_{12} \sim U_0$ の出力 $u_{12} \sim u_1, h_0$ の全てが論理 “1” となり、残余のORゲート U_{14}, U_{13} の出力 u_{14}, u_{13} は論理 “0” となる。

【0106】

このようにORゲート $U_{14} \sim U_0$ は、入力データBの各ビット $b_{15} \sim b_0$ のうちで、論理 “1” となっている最上位のビットを検出すると共に、その論理 “1” となっているビット及びそのビットより下位に位置しているORゲートの出力を全て論理 “1” とすることで、上述したアクティブビットの候補を出力する。

【0107】

次に、ANDゲート $G_{14} \sim G_0$ の動作を説明する。

【0108】

ANDゲート G_{14} は、図示の如く、ビット b_{15}, b_{14} との論理積演算を行い、ANDゲート $G_{13} \sim G_0$ は、ORゲート $U_{14} \sim U_1$ よりの出力 $u_{14} \sim u_1$ とビット $b_{13} \sim b_0$ との夫々の論理積演算を行う。そして、デコードビット g_{15} と、これらのANDゲート $G_{14} \sim G_0$ の出力（デコードビット） $g_{14} \sim g_0$ とによって、真のアクティブビット（すなわち、1つのアクティブビット）及びそのビット番号Sを示す検出結果を出力する。

【0109】

つまり、入力データBの最上位ビット b_{15} が論理 “1” のときには、デコードビット g_{15} が論理 “1” となり、ANDゲート G_{14} から出力されるデコードビット g_{14} は論理 “0” となる。

【0110】

更に、ORゲートU₁₄の出力u₁₄はビットb₁₅が論理“1”であるため論理“1”となり、更に残余のORゲートU₁₃～U₁の出力u₁₃～u₁も論理“1”となる。そして、ANDゲートG₁₃～G₀の論理反転入力端子に、全てが論理“1”となっている出力u₁₃～u₁が印加される結果、ANDゲートG₁₃～G₀から出力されるデコードビットg₁₃～g₀は全て論理“0”になる。

【0111】

このようにORゲートU₁₄～U₁とANDゲートG₁₄～G₀とによる論理演算が行われると、デコードビットg₁₅～g₀のうち、ビットg₁₅のみが論理“1”、残余のビットg₁₄～g₀は全て論理“0”となり、上述した真のアクティブビットb₁₅及びそのビット番号S（すなわち、15）を検出して特定する。

【0112】

また、入力データBの第15ビットb₁₅が論理“0”で、第14ビットb₁₄が論理“1”的ときには、デコードビットg₁₅が論理“0”となり、ANDゲートG₁₄から出力されるデコードビットg₁₄は論理“1”となる。

【0113】

更に、ORゲートU₁₄～U₁の出力u₁₄～u₁は全て論理“1”になるため、ANDゲートG₁₃～G₀のデコードビットg₁₃～g₀は全て論理“1”となる。

【0114】

したがって、入力データBの第14ビットb₁₄が論理“1”となる最上位のビットであった場合には、デコードビットg₁₅～g₀のうちのビットg₁₄のみが論理“1”となって、第14ビットb₁₄を真のアクティブビットとし更にそのビット番号S（すなわち、14）を検出して特定する。

【0115】

また、入力データBの第15ビットb₁₅と第14ビットb₁₄が共に論理“0”で、第13ビットb₁₃が論理“1”的ときには、デコードビットg₁₅、g₁₄が共に論理“0”となる。

【0116】

更に、ORゲートU₁₄の出力u₁₄は論理“0”となり、第13ビットb₁₃は論理“1”であるから、ANDゲートG₁₃のデコードビットg₁₃は論理“1”となる。

【0117】

そして更に、ORゲートU₁₃～U₁の出力u₁₃～u₁は全て論理“1”になるため、ANDゲートG₁₂～G₀のデコードビットg₁₂～g₀は全て論理“0”となる。

【0118】

したがって、入力データBの第13ビットb₁₃が論理“1”となる最上位のビットであった場合には、デコードビットg₁₅～g₀のうちのビットg₁₃のみが論理“1”となって、第13ビットb₁₃を真のアクティブビットとし更にそのビット番号S（すなわち、13）を検出して特定する。

【0119】

以下同様の論理演算処理が行われ、デコードビットg₁₅～g₀のうちの1つのデコードビットのみが論理“1”となることから、その論理“1”となったデコードビットに対応している入力データBのビットをアクティブビットとして検出すると共に、そのビット番号Sを一義的に特定することが可能となっている。

【0120】

次に、マルチプレクサ13の動作を説明する。マルチプレクサ13は、既述した如く、17個の制御端子eのうち、論理“1”が印加された制御端子eに対応している入力端子dのバーナリデータを、ビット列(d₆, d₅, d₄)として出力ポートより出力する。

【0121】

したがって、デコードビットg₁₅～g₀のうちの1つのデコードビットのみが論理“1”となると、その論理“1”が印加される制御端子eに対応している入力端子dのバーナリデータを、ビット列(d₆, d₅, d₄)として出力する。

【0122】

その結果、入力データBのうちのアクティブビットに相当するビット列（ d_6 , d_5 , d_4 ）を出力する。

【0123】

ここで、夫々の入力端子dには、ビット番号Sに相当する2進数のバイナリデータが予め印加されているため、デコードビット $g_{15} \sim g_0$ で示されるアクティブビットのビット番号Sに相当するバイナリデータをビット列（ d_6 , d_5 , d_4 ）として出力する。

【0124】

例えば、入力データB（ b_{15}, \dots, b_0 ）の各ビット列が上述した（0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1）であった場合、図6中のデコードビット $g_{15} \sim g_0$ のうちのビット g_{12} のみが論理“1”となり、ビット g_{12} が印加される制御端子eに対応している入力端子dのバイナリデータは、（1, 0, 0）であるため、そのバイナリデータ（1, 0, 0）をビット列（ d_6 , d_5 , d_4 ）として出力する。

【0125】

更に、ORゲートU₈の出力 u_8 を最上位のビット d_7 として、総計4ビットから成るビット列（ d_7, d_6, d_5, d_4 ）を対数変換上位ビット列D_{UP}（ d_7, d_6, d_5, d_4 ）、すなわちD_{UP}（1, 1, 0, 0）とする。

【0126】

なお、入力データBの第15ビットから第7ビットまでのビット列（ $b_{15} \sim b_7$ ）の何れかのビットが論理“1”となると、ORゲートU₈の出力 u_8 は必ず論理“1”となり、そのため対数変換上位ビット列D_{UP}（ d_7, d_6, d_5, d_4 ）のうちのビット d_7 は論理“1”となる。

【0127】

また、入力データBの第15ビットから第7ビットまでのビット列（ $b_{15} \sim b_7$ ）の全てのビットが論理“0”となると、ORゲートU₈の出力 u_8 は必ず論理“0”となり、そのため対数変換上位ビット列D_{UP}（ d_7, d_6, d_5, d_4 ）のうちのビット d_7 は論理“0”となる。

【0128】

このように、入力データBの第15ビットから第7ビットまでのビット列（ $b_{15} \sim b_7$ ）の何れか1つが論理“1”となったか、全てが論理“0”となったかに応じて、対数変換上位ビット列 D_{UP} （ d_7, d_6, d_5, d_4 ）のビット d_7 を論理“1”又は“0”に設定することにより、アクティブビットのビット番号Sに相当する対数変換上位ビット列 D_{UP} を生成することを可能にしている。

【0129】

図8及び図9は、以上に述べた対数変換上位ビット列生成部3の動作を真理値表として示した図である。なお、紙面の都合上、入力データBの代表的な値に対する対数変換上位ビット列 D_{UP} の出力結果を示している。また、入力データBは10進数と2進数で示し、対数変換上位ビット列 D_{UP} の値を10進数で示している。

【0130】

図8に示すように、様々な値を示す入力データBが入力されると、その入力データBの各ビットのうち、論理“1”となる最も上位のビット（アクティブビット）のビット番号Sに相当する対数変換上位ビット列 D_{UP} を出力する。したがって、対数変換上位ビット列 D_{UP} の値を10進数で表すと、図示の如く、入力データBを対数演算した場合に得られる整数値を対数変換上位ビット列 D_{UP} として出力するようになっている。

【0131】

つまり、図9中、□で囲って示されているアクティブビットよりも下位のビット（符号「*」で示している）をドントケア（Don't Care）として、アクティブビットのビット番号Sを対数変換し、それによって得られる整数の対数値を対数変換上位ビット列 D_{UP} として出力するようになっている。

【0132】

次に、図7を参照して、本実施例の対数変換下位ビット列生成部4の構成を説明する。

この対数変換下位ビット列生成部4は、複数のANDゲート $X_{15} \sim X_1$ と、ORゲート X_0 と、マルチプレクサ14を備えて構成されている。

【0133】

ここで、マルチプレクサ14は、図6に示したマルチプレクサ13と同様の構成を有しており、16個の入力端子dを備えるデータ入力ポートと、夫々の入力端子dに対応して1組ずつの対を成している16個の制御入力端子eを備える切換え制御ポートが設けられ、更に、後述する4ビット($m=4$)の対数変換下位ビット列 D_{LOW} (d_3, d_2, d_1, d_0)を出力する出力ポートが設けられている。

【0134】

そして、16個の制御端子eのうちの1つの制御端子が論理“1”に設定されると、その制御端子と対をなしている入力端子dに印加されている4ビットのバイナリデータを対数変換下位ビット列 D_{LOW} (d_3, d_2, d_1, d_0)として出力する。

【0135】

ここで、図示するように、マルチプレクサ14の最上位に位置している制御端子eとANDゲート X_{15} の出力とが接続され、次に位置している制御端子eとANDゲート X_{14} の出力とが接続され、更に次に位置している制御端子eとANDゲート X_{13} の出力とが接続され、以下同様に、残余の制御端子eとANDゲート $X_{12} \sim X_1$ の夫々の出力とが一対一に対応付けて接続され、最下位に位置している制御端子eには、ORゲート X_0 の出力とが接続されている。

【0136】

更に、上述した最上位から最下位に位置している16個の各制御端子eに対応して設けられている各入力端子dには、入力データB($d_{15}, d_{14}, \dots, b_1, b_0$)のうちのビット $d_{15}, d_{14}, \dots, b_1, b_0$ が、4ビットずつずらした関係で割り当てて供給されている。

【0137】

つまり、ANDゲート X_{15} に接続されている制御端子eと対を成している最上位の入力端子dにはビット列($d_{14}, d_{13}, d_{12}, d_{11}$)が並列入力され、ANDゲート X_{14} に接続されている制御端子eと対を成している最上位の入力端子dにはビット列($d_{13}, d_{12}, d_{11}, d_{10}$)が並列入力され

、ANDゲートX₁₃に接続されている制御端子eと対を成している最上位の入力端子dにはビット列(d₁₂, d₁₁, d₁₀, d₉)が並列入力され、以下同様にして、残余の入力端子dにも、入力データBのうちの特定の4ビットが順次に並列入力されている。

【0138】

ただし、このように入力データBを4ビットずつずらした関係で割り当てていくと、割り当てるべきビットが途中で不足することになり、それに対処すべく、不足分のビットを論理“0”のビットにして追加することで、入力端子dには必ず4ビットのバイナリデータが入力されるようになっている。

【0139】

つまり、マルチプレクサ14の最下位に位置している入力端子(ORゲートX₀と接続している制御端子eと対をなしている入力端子)dには、バイナリーデータ(0, 0, 0, 0)が印加され、その上に位置している入力端子dには、バイナリーデータ(b₀, 0, 0, 0)が印加され、更にその上に位置している2つの入力端子d, dには、夫々バイナリーデータ(b₁, b₀, 0, 0)と(b₂, b₁, b₀, 0)が印加されている。

【0140】

また、追加した論理“0”的ビットは、いわゆるグランド端子にプルダウン抵抗を介して接続するという簡単な回路構成によって設定されている。

【0141】

次に、ANDゲートX₁₅～X₁は、4入力型のANDゲートで形成され、対数変換上位ビット列D_{UP}(d₇, d₆, d₅, d₄)を入力すべく配線がなされている。

【0142】

また、図示のように、ANDゲートX₁₅～X₁の4つの入力端子のうち、所定の入力端子が論理反転入力端子となっている。

【0143】

したがって、ANDゲートX₁₅～X₁は、対数変換上位ビット列D_{UP}(d₇, d₆, d₅, d₄)のバイナリの値を検出し、その結果、上述したアクティ

ブピットのビット番号 S を検出する。

【0144】

具体的な場合を例示すれば、前述した入力データ B (0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1) が入力され、対数変換上位ビット列 D_{UP} (d₇, d₆, d₅, d₄) として、D_{UP} (1, 1, 0, 0) が入力されると、ANDゲート X₁₂ の出力のみが論理 “1” となり、残余のANDゲート X₁₅～X₁₃ と X₁₁～X₁ の出力は全て論理 “0” になる。

【0145】

したがって、ANDゲート X₁₅～X₁ によって、対数変換上位ビット列 D_{UP} (1, 1, 0, 0) に相当する第12ビット b₁₂ をアクティブビットとして検出し、ANDゲート X₁₂ の論理 “1” の出力をマルチプレクサ 14 の所定の制御端子 e に供給する。

【0146】

なお、ANDゲート X₁₂ に接続されている当該制御端子 e に対応している入力端子 e には、入力データ B の第11ビットから第8ビットのビット列 (b₁₁～b₈) がバイナリデータとして印加されているため、上述の入力データ B (0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1) のうちのビット列 (0, 1, 1, 0) が、対数変換下位ビット列 D_{LOW} (d₃, d₂, d₁, d₀) としてマルチプレクサ 14 より出力される。

【0147】

このように、ANDゲート X₁₅～X₁ は、対数変換上位ビット列 D_{UP} (d₇, d₆, d₅, d₄) で示されるアクティブビットのビット番号 S を検出し、更に、入力データ B のビット列のうち、ビット番号 S よりも下位4ビット分のビット列のデータをマルチプレクサ 14 より出力させるようになっている。

【0148】

次に、図10及び図11を参照して対数変換下位ビット列生成部4の動作を説明する。なお、図10及び図11は、動作を真理値表として示した図であり、入力データ B は10進数と2進数で示し、最終的に生成される対数変換データ D の値を10進数で示している。

【0149】

まず図9において、入力データBの各ビットのうち、同図中の口で囲まれているアクティブビットに基づいて、対数変換上位ビット列生成部3が対数変換上位ビット列D_{UP}を生成する。

【0150】

これに対して、対数変換下位ビット列生成部4は、図9中の符号「*」で示されているアクティブビットよりも下位の4ビット分を、小数点以下の値を表す対数変換下位ビット列D_{LOW}とする。

【0151】

したがって、図10に示すように、図9中の符号「*」で示されているビット列を全体的に1ビット分下位側にシフトさせて、符号「・」で示す位置を小数点とすると、符号「・」を境にして、整数の対数変換値を表す対数変換上位ビット列D_{UP}と、小数点以下の対数変換値を表す対数変換下位ビット列D_{LOW}とかからなる対数変換データDが生成される。

【0152】

対数変換下位ビット列生成部4は、図7に示した回路によって、この小数点以下の対数変換値を表す対数変換下位ビット列D_{LOW}を入力データBから選択して抽出する処理を行い、その対数変換下位ビット列D_{LOW}を対数変換上位ビット列D_{UP}に続く下位のビット列として出力する。

【0153】

つまり、図11に示すように、口で囲まれているアクティブビットに基づいて生成された対数変換上位ビット列D_{UP}に続けて、そのアクティブビットよりも下位4ビット分のビット列をそのまま対数変換下位ビット列D_{LOW}として出力する。

【0154】

そして、本対数変換器が、対数変換上位ビット列D_{UP}と対数変換下位ビット列D_{LOW}とを合わせたmビットのバイナリデータを対数変換データDとして出力する。

【0155】

この結果、対数変換データDは、図11中に10進数の値として示されているように、整数値と小数点以下の値との和として、対数変換の結果を表すこととなり、十分な精度が得られない整数値に対して小数点以下の値が加わることで、より精度の高い対数変換データDを生成することを可能にしている。

【0156】

図12は、本実施例の対数変換器の変換精度を示す図であり、16ビットの入力データBを8ビットの対数変換データDに対数変換した場合の評価結果を示している。

【0157】

同図から解る通り、入力データBを10進数の「0」から「655.35」まで変化させ、夫々の値に対応する対数出力の理論値と、本対数変換器により求めた対数変換データDの実際の結果を比較すると、ほぼ理論値に極めて近い結果が得られた。これにより、極めて高い対数変換精度が得られることを実証することができた。

【0158】

以上説明したように、本実施形態と実施例及びそれらの変形例によれば、少ない論理演算素子で対数変換を行うことができるため、回路規模の大幅な低減等を実現することができる。また、対数変換に要する処理時間を大幅に短縮することも可能である。また、図12に示したように、高い対数変換精度を得ることができる。

【0159】

このように、対数変換器に要求される要件を満足した新規な構成の対数変換器を提供することができ、デジタル信号処理を行う各分野、例えばマルチメディア化に伴って研究開発が進められているデジタル通信機器やデジタルテレビジョンセット等を含めた広い分野に、有効な効果を提供することができる。

【0160】

なお、図2、図4、図7、図9、図10、図11を参照して説明したように、本実施形態と実施例及びそれらの変形例では、小数点以下の対数変換値を表す2進数のバイナリデータ列（対数変換下位ビット列） D_{LOW} を、入力データBの

アクティブビットより下位に位置するビット列から抽出して、対数変換上位ビット列 D_{UP} と合わせることで、最終的に対数変換データ D を生成することとしている。

【0161】

そして、入力データ B のアクティブビットより下位に位置するビット列から抽出する処理を1回だけ行って、対数変換上位ビット列 D_{UP} と合わせるだけで、対数変換データ D を生成することとしているので、対数変換の処理に要する時間を大幅に短縮するのとが可能となっている。

【0162】

ただし、更に対数変換精度を向上させるべく次の処理を行うようにしてもよい。

【0163】

つまり、入力データ B のアクティブビットより下位に位置するビット列、つまり、予め決められた q 個 ($q = 4$) のビット列ではなく、アクティブビットより下位に位置する残余の全てのビット列の中で、上述したの同様の論理“1”となる最も上位に位置するビットを検出する。すなわち、図9、図10中の符号「*」で示されている小数点以下の対数値を表すビット列の中から、アクティブビット（以下「小数点以下アクティブビット」という）を検出する。

【0164】

そして、小数点以下アクティブビットの位置に相当する2進数のバイナリーデータ（小数点以下の値を表すバイナリーデータ）と、その小数点以下アクティブビットより下位の残りのビット列によって、小数点以下の対数値を表す対数変換下位ビット列 D_{LOW} を生成する。

【0165】

このようにすると、限られたビット数 m の対数変換データ D を生成する一般的要請に応じるべく、小数点以下の対数値を表すバイナリーデータのビット数を低減することとなっても、精密に小数点以下の対数値を表す対数変換データ D を生成することができる。別言すれば、少ないビット数で、小数点以下の対数値の分解能を向上させることができ、それによって、入力データ B に対応する高精度の

対数変換データDを生成することができる。

【0166】

更に、こうした小数点以下アクティビットを検出する処理を複数回（2回以上）繰り返してもよい。2回以上繰り返すと、小数点以下の対数値の分解能を更に向上させることができ、高精度の対数変換データDmを求めることが可能となる。

【0167】

また、実施形態及び実施例の説明では、主にハードウェアで形成される対数変換器について説明したが、デジタル演算処理が可能なコンピュータ（C P U）等を備えた電子機器で実行される対数変換用のコンピュータプログラムとすることも可能である。

【0168】

つまり、実施形態及び実施例で説明した対数変換器と同等の機能を有するコンピュータプログラムを作成し、パーソナルコンピュータ（P C）等の電子機器で実行させることにより、対数変換に要する処理時間を大幅に短縮化し、また高い対数変換精度等を得ることができる。

【図面の簡単な説明】

【図1】

従来の対数変換器の構成を示す図である。

【図2】

本発明の実施の形態に係る対数変換器の構成及び作用を説明するための図である。

【図3】

本発明の実施の形態に係る対数変換器の構成及び作用を更に説明するための図である。

【図4】

本発明の実施の形態に係る対数変換器の構成及び作用を更に説明するための図である。

【図5】

本実施例の対数変換器を適用した電子機器の構成を示す図である。

【図6】

本実施例の対数変換器に備えられた回路の構成を示す図である。

【図7】

本実施例の対数変換器に備えられた回路の構成を更に示した図である。

【図8】

本実施例の対数変換器の動作を説明するための図である。

【図9】

本実施例の対数変換器の動作を更に説明するための図である。

【図10】

本実施例の対数変換器の動作を更に説明するための図である。

【図11】

本実施例の対数変換器の動作を更に説明するための図である。

【図12】

本実施例の対数変換器における変換精度の評価結果を示す図である。

【符号の説明】

3 …対数変換上位ビット列生成部

3 a …アクティブビット検出部

3 b …上位ビット列生成部

4 …対数変換下位ビット列生成部

4 a …下位ビット選択部

4 b …上位ビット列抽出部

1 2 …AM検波部

1 3, 1 4 …マルチプレクサ

U₀ ~ U₁₄ …ORゲート

G₀ ~ G₁₄ …ANDゲート

X₀ …ORゲート

X₁ ~ X₁₅ …ANDゲート

e …制御端子

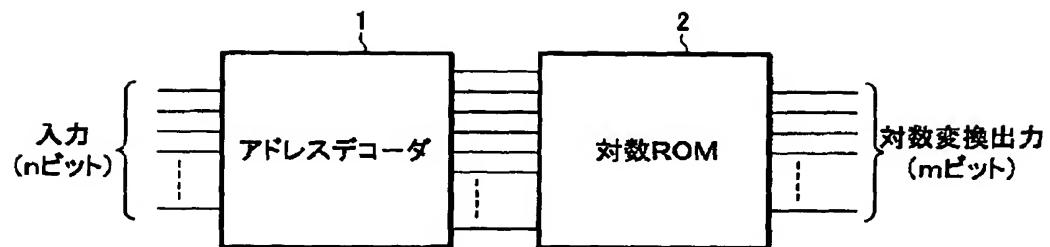
特2002-214046

d …入力端子

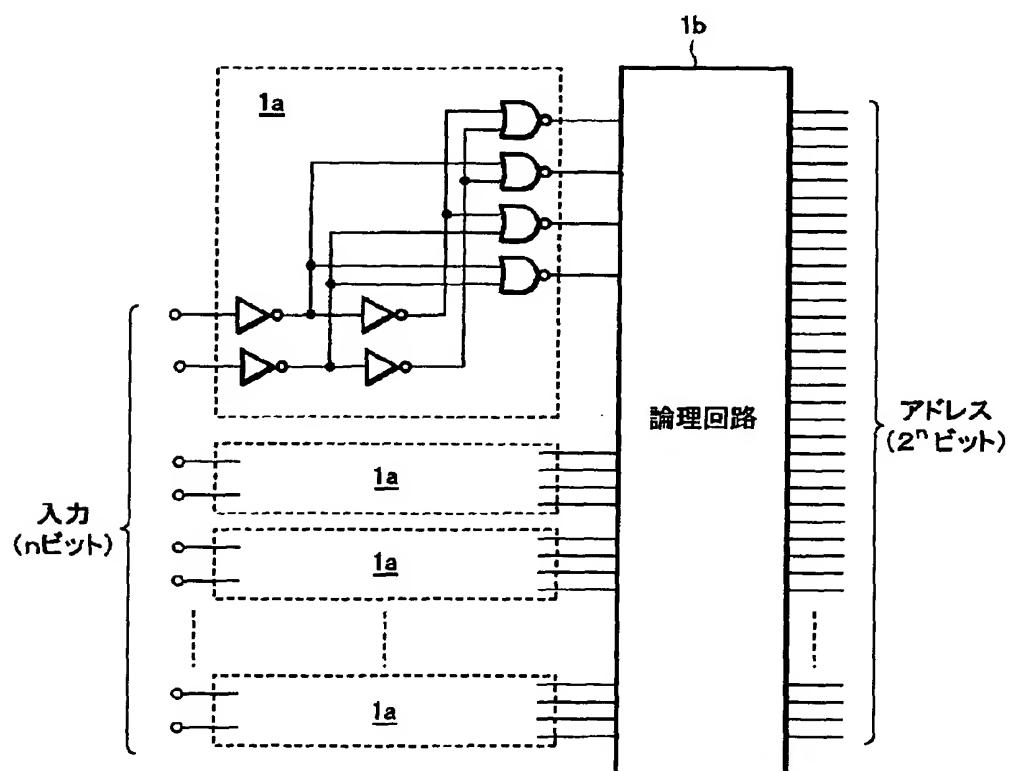
【書類名】 図面

【図1】

(a)

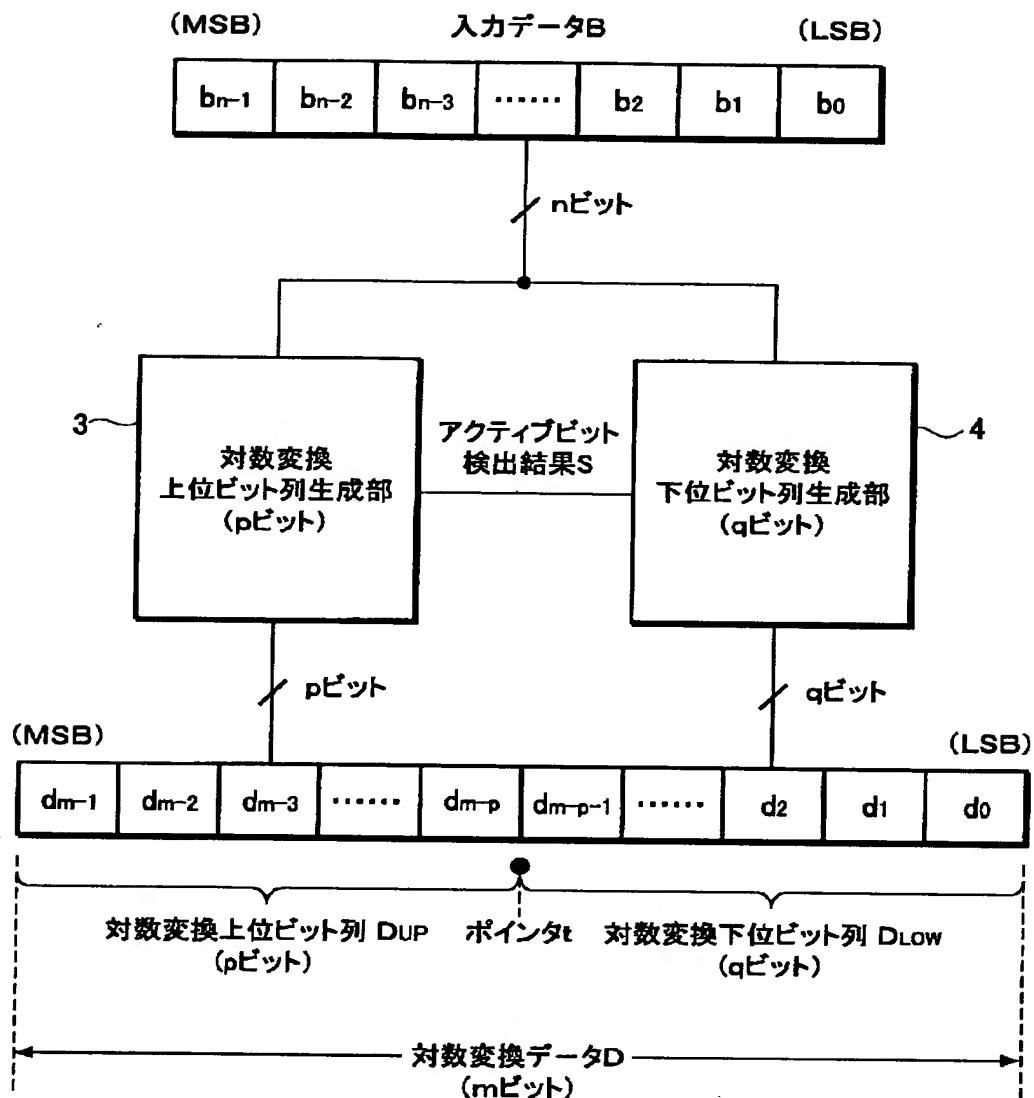


(b)

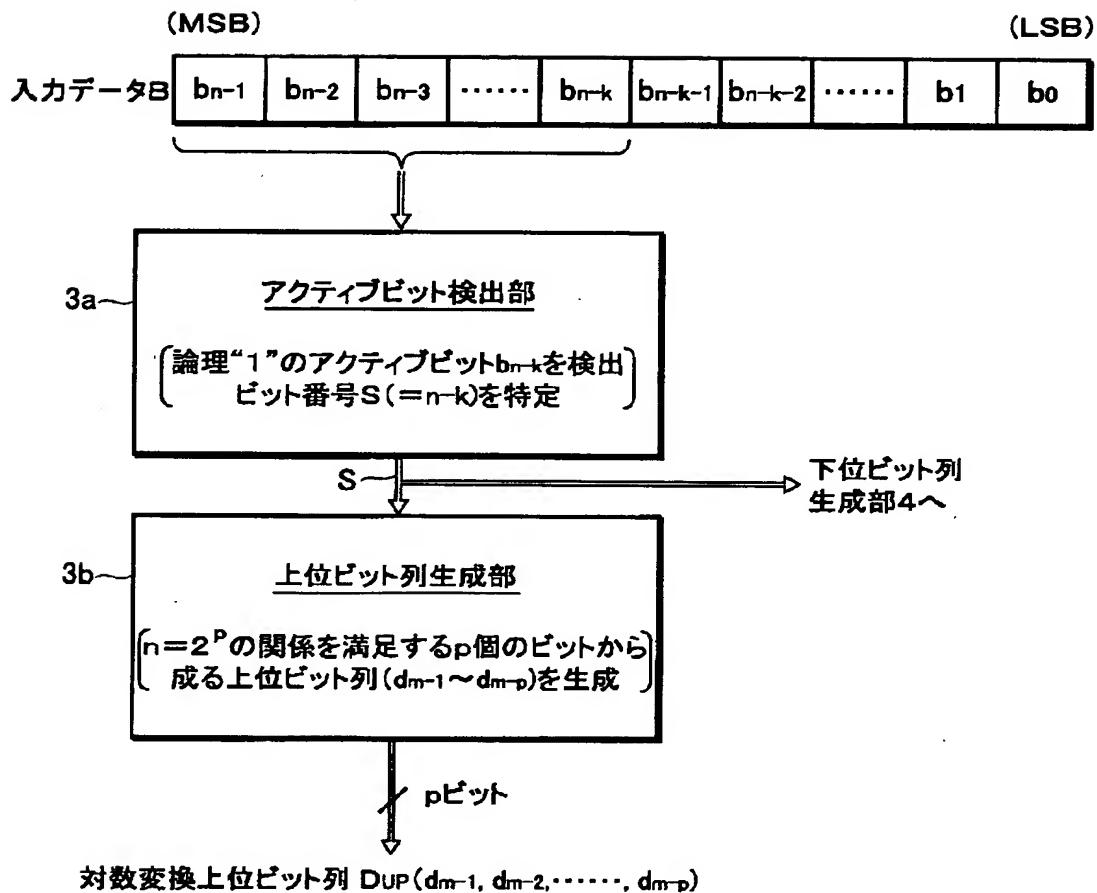


【従来技術】

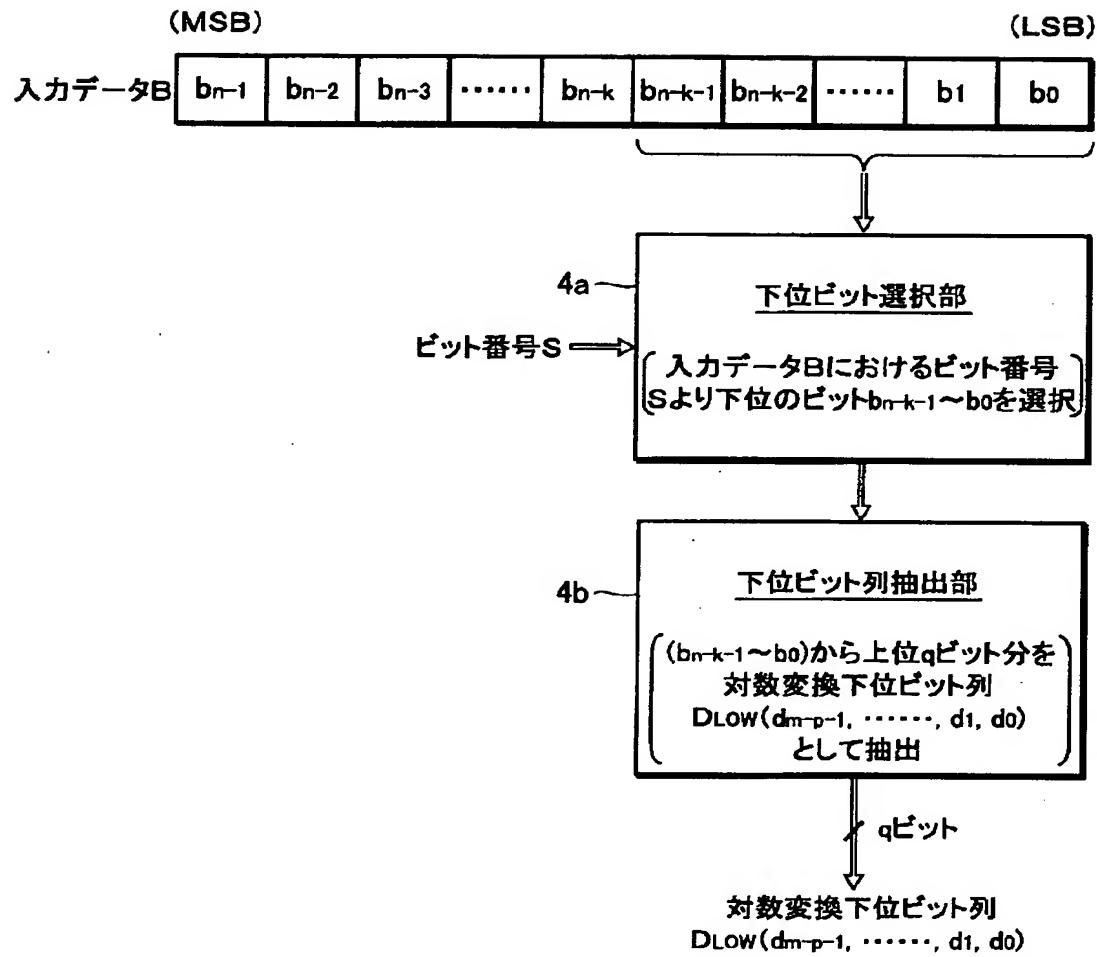
【図2】



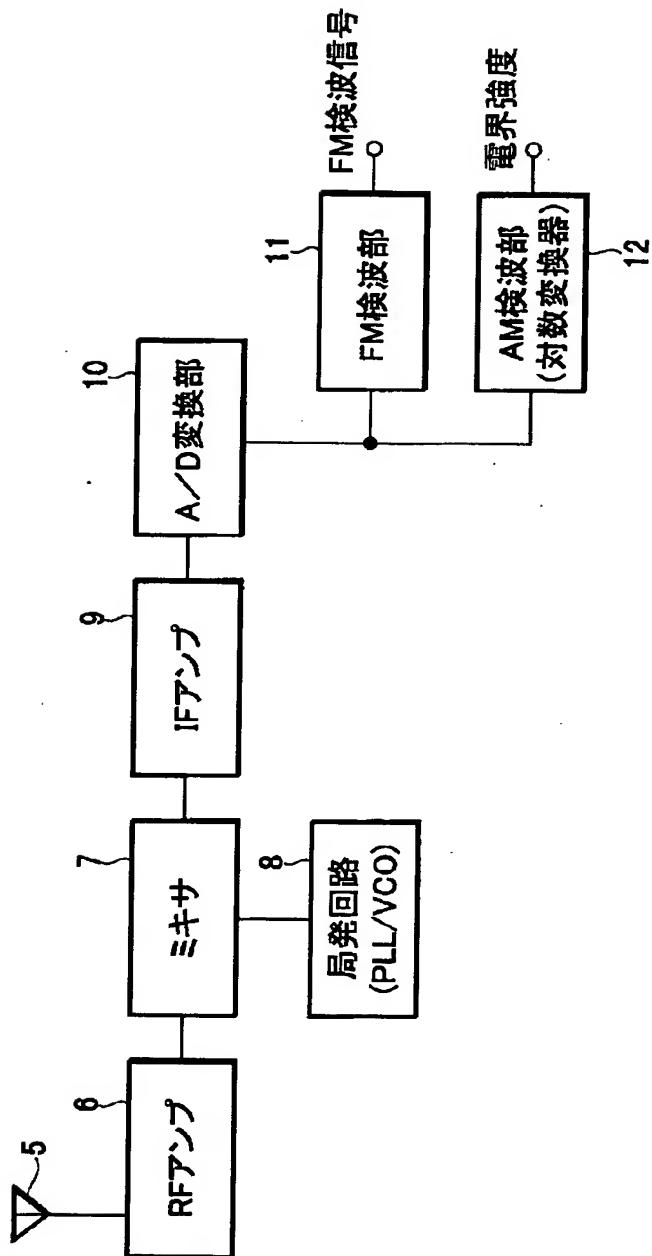
【図3】



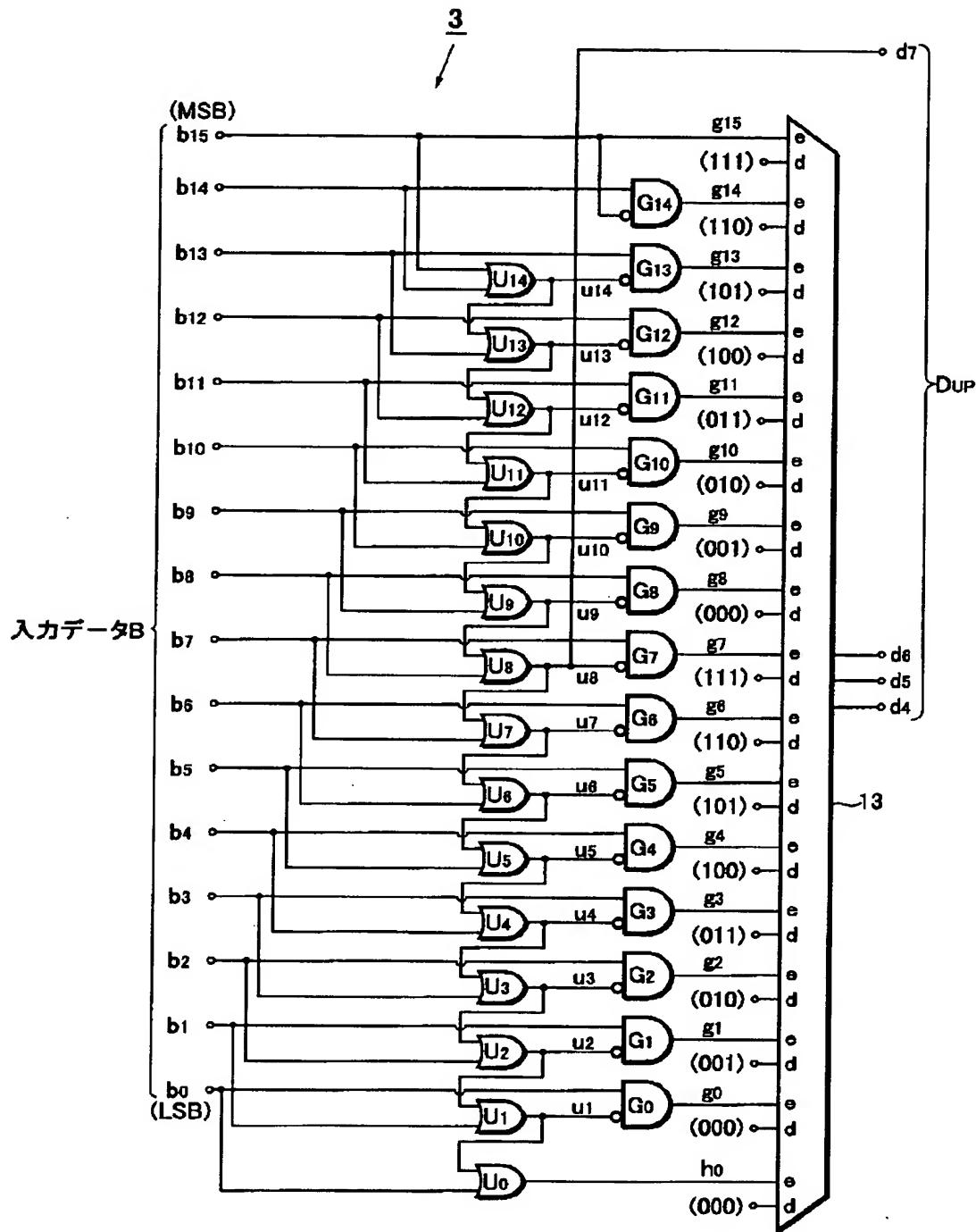
【図4】



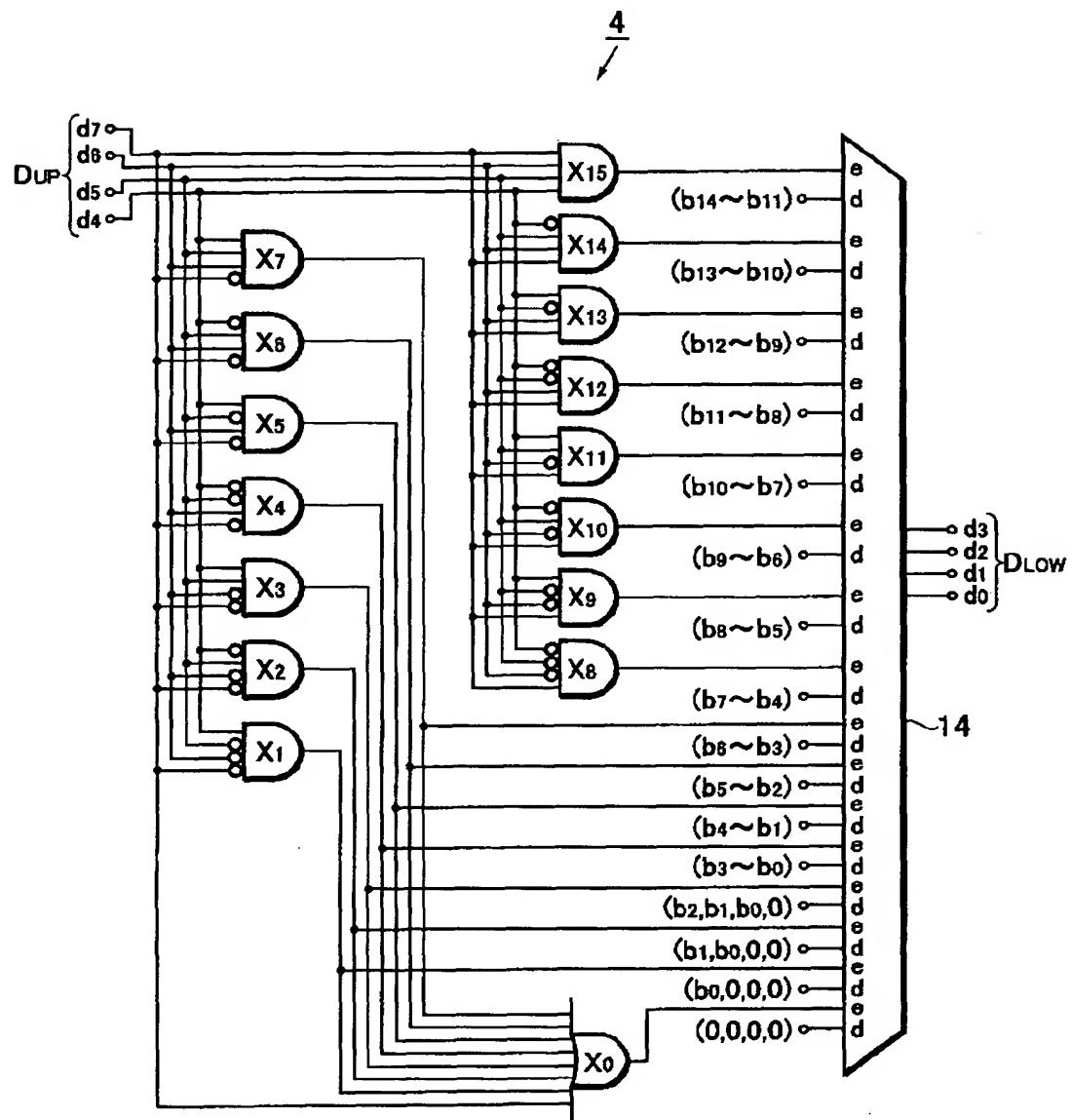
【図5】



【図6】



【図7】



【図8】

入力(16ビットの場合)			対数変換上位ビット列D _{UP} 10進数
10進数	2進数(バイナリ) (MSB)	(LSB)	
0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	1
2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	2
3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1	3
4	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	4
5	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	5
6	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0	6
7	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1	7
8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	8
9	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	9
10	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	10
11	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1	11
12	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0	12
13	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0	13
14	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0	14
15	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1	15
16	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	4
32	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	5
64	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	6
128	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	7
256	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	8
512	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	9
1024	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	10
2048	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11
4096	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	12
8192	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	13
16384	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	14
32768	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15
65535	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	

【図9】

入力 (16ビットの場合)			対数変換上位ビット列D↑P 10進数
10進数	2進数 (バイナリ)		
	(MSB)	(LSB)	
0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1		1
2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 *		2
4	0 0 0 0 0 0 0 0 0 0 0 0 0 1 *		3
8	0 0 0 0 0 0 0 0 0 0 0 0 1 *		4
16	0 0 0 0 0 0 0 0 0 0 0 1 *		5
32	0 0 0 0 0 0 0 0 0 1 *		6
64	0 0 0 0 0 0 0 0 1 *		7
128	0 0 0 0 0 0 0 1 *		8
256	0 0 0 0 0 0 1 *		9
512	0 0 0 0 0 1 *		10
1024	0 0 0 0 1 *		11
2048	0 0 0 1 *		12
4096	0 0 1 *		13
8192	0 1 *		14
16384	0 1 *		15
32768	1 *		

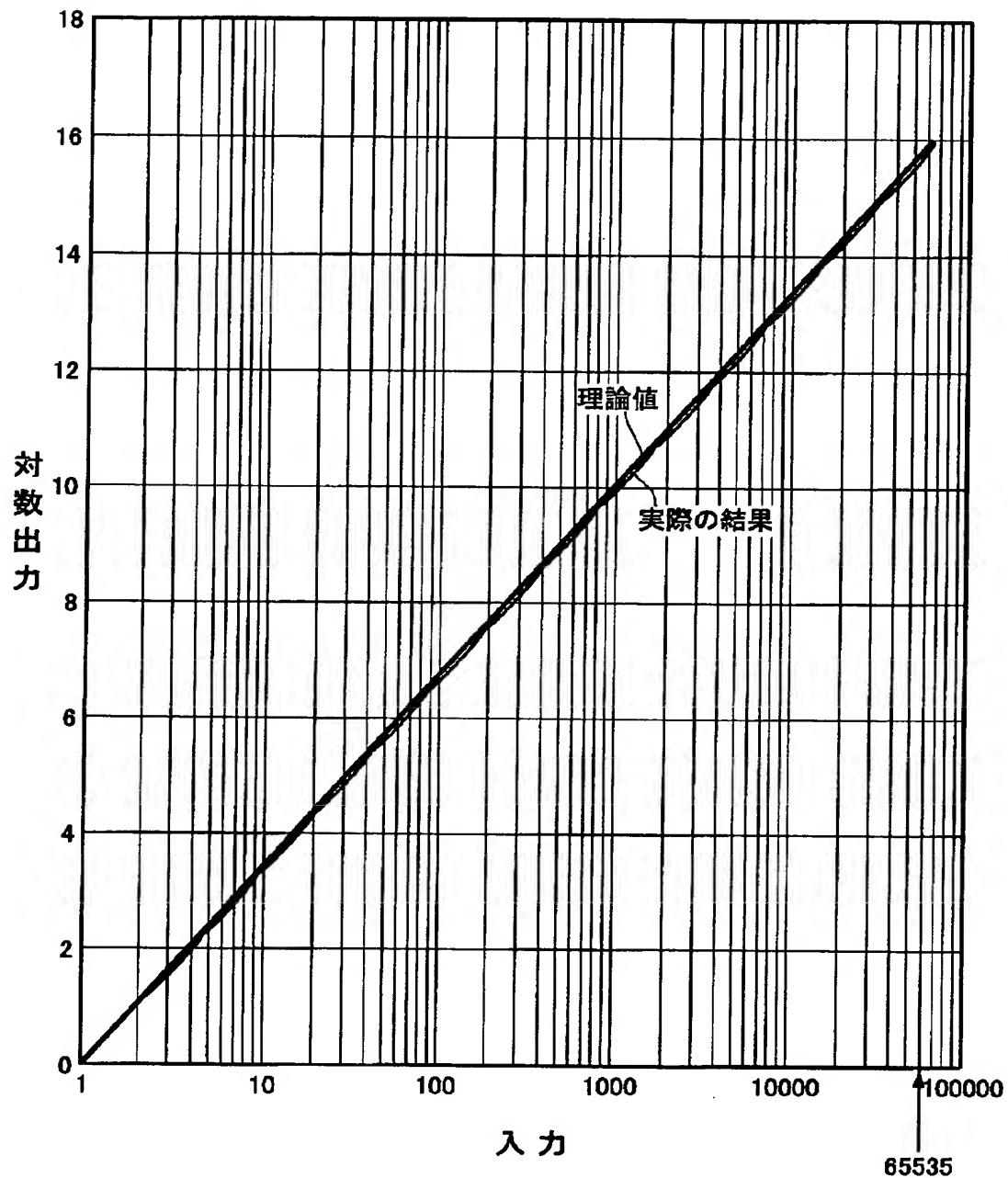
【図10】

入力 (16ビットの場合)			対数変換上位ビット列D 10進数
10進数	2進数 (バイナリ)		
	(MSB)	(LSB)	
0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		0
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1		1
2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 *		2
4	0 0 0 0 0 0 0 0 0 0 0 0 0 1 *		3
8	0 0 0 0 0 0 0 0 0 0 0 0 1 *		4
16	0 0 0 0 0 0 0 0 0 0 0 1 *		5
32	0 0 0 0 0 0 0 0 0 0 1 *		6
64	0 0 0 0 0 0 0 0 0 1 *		7
128	0 0 0 0 0 0 0 0 1 *		8
256	0 0 0 0 0 0 0 1 *		9
512	0 0 0 0 0 0 1 *		10
1024	0 0 0 0 0 1 *		11
2048	0 0 0 0 1 *		12
4096	0 0 0 1 *		13
8192	0 0 1 *		14
16384	0 1 *		15
32768	1 *		

【図11】

入力 (16ビットの場合)			対数変換データ D ($\log_2 B$) 10進数
10進数	2進数 (バイナリ) (MSB)	(LSB)	
0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0.0000
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	1.0000
2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0	1.5000
3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 1 1	2.0000
4	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	2.2500
5	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 1 0 1	2.5000
6	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 1 1 0	2.7500
7	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 1 1 1	3.0000
8	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 1 0 0 0	3.1250
9	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 1 0 0 0 1	3.2500
10	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0 1 0 1 0	3.3750
11	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 0 1 0 1 1	3.5000
12	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1 1 0 0 0	3.6250
13	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1 1 0 1	3.7500
14	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1 1 1 0	3.8750
15	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 1 1 1 1 1	4.0000
16	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1	0 0 0 0 0 1 0 0 0 0 0 0	
32	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	5.0000
64	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	6.0000
128	0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	7.0000
256	0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	8.0000
512	0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	9.0000
1024	0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	10.0000
2048	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	11.0000
4096	0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	12.0000
8192	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	13.0000
16384	0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	14.0000
32768	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15.0000
36864	1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15.1250
40960	1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15.2500
45056	1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15.3750
49152	1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15.5000
53248	1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15.6250
57344	1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15.7500
61440	1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	15.8750

【図12】



【書類名】 要約書

【要約】

【課題】 回路規模の低減が可能な対数変換器を提供する。

【解決手段】 対数変換上位ビット列生成部3が、入力データBの各ビット $b_{n-1} \cdots b_0$ のうち論理“1”となっている最も上位に位置するビットをアクティブビットとして検出し、アクティブビットのビット位置Sを表すバイナリデータを対数変換上位ビット列 $D_{UP}(d_{m-1} \cdots d_{m-p})$ とする。ここで、入力データBのビット数nに基いて、 $n = 2^p$ の関係から対数変換上位ビット列 $D_{UP}(d_{m-1} \cdots d_{m-p})$ のビット数pを設定する。対数変換下位ビット列生成手段4が、入力データBの各ビット $b_{n-1} \cdots b_0$ のうち、ビット位置Sより下位に位置する所定ビット数qのビット列を求め、対数変換下位ビット列 $D_{LOW}(d_{m-p-1} \cdots d_0)$ とする。そして、対数変換上位ビット列 $D_{UP}(d_{m-1} \cdots d_{m-p})$ を入力データBを対数変換した場合に得られる整数の対数変換値、対数変換下位ビット列 $D_{LOW}(d_{m-p-1} \cdots d_0)$ を入力データBを対数変換した場合に得られる小数点以下の対数変換値とする総計 $p+q$ ビットの対数変換データDを生成し、対数変換した結果として出力する。

【選択図】 図2

出願人履歴情報

識別番号 [000005016]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都目黒区目黒1丁目4番1号

氏 名 パイオニア株式会社